High-Speed Wide-Area Distributed Data Handling: Architecture, Implementation, and Issues[1]

Brian Tierney, and Craig Tull, NERSC Division
William E. Johnston, Information and Computing Sciences Division
Douglas Olson, Nuclear Science
Ernest Orlando Lawrence Berkeley National Laboratory
University of California

1. This work is supported by the Director, Office of Energy Research, Office of Computation and Technology Research, Mathematical, Information, and Computational Sciences Division, of the U. S. Department of Energy under Contract No. DE-AC03-76SF00098 with the University of California, and by DARPA ISTO. E-mail: bltierney@lbl.gov, wejohnston@lbl.gov

Data-Intensive Computing in Widely Distributed Environments

The overall goal of this work is to provide methodology and tools to enable the scientific community to routinely deal with massive volumes of data at high data rates with complete location transparency.

Wide Area, Data Intensive Computing

Challenge

Identify what must be done to produce

- **predictable, high-speed, distributed software components that will compose to yield high-performance widely distributed applications .....**
  (rather than having to "tune" these systems from top-to-bottom as we mostly have to do now)

Increasingly, we believe that meeting this challenge will involve

- **comprehensive and adaptable monitoring that provides the information that system components need in order to adapt to changes in the communications environment, and**
- **automated distributed management of the distributed components**

Wide Area, Data Intensive Computing

Issues and Approaches

- **Scalability - the system must grow and shrink "gracefully"**
  - **dynamic (re-) configuration**
- **Performance in high-speed, but variable quality, networks**
  - **network-level and platform-level parallelism and pipelining**
  - **individual threads of control for every server-level resource**
  - **monitoring at all levels for problem diagnosis, algorithm analysis, and application adaptation**
- **Reliability (otherwise the system is not real)**
  - **autonomous monitoring and management for**
    - **system regeneration in the event of transient failure,**
    - **system reconfiguration in the event of long-term failure, and**
    - **adaptability in the event of degradation**

Wide Area, Data Intensive Computing

- **Security (otherwise the system is not real)**
  - **decentralized management of access rights to match the decentralized nature of the distributed systems: PKI certificate based distributed management of policy**

Wide Area, Data Intensive Computing

Outline of What Follows

- **Overall model**
- **Prototype systems**
  - **WALDO: A digital library for real-time data sources**
  - **STAR: Distributed physics/nuclear science data analysis**
- **DPSS: A wide-area network distributed cache**
- **Performance monitoring and analysis**
  - **Event-based methodology at all levels**
  - **The MAGIC WAN experiment**
  - **Application monitoring example: The STAR analysis framework**
- ~~**Agent-based monitoring**~~
- ~~**Security**~~
- **The LBNL-SLAC-NTON experiment**

An Overall Model for Data-Intensive Computing

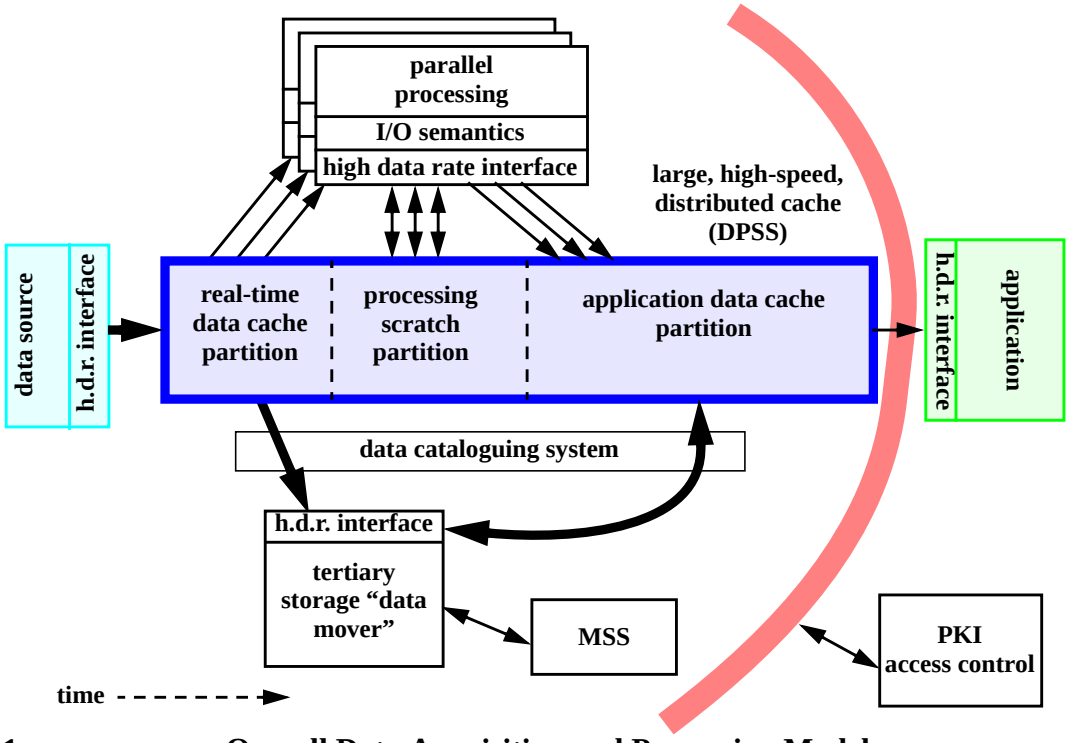
- **Data sources deposit data in a distributed cache, and consumers takes data from the cache, usually writing processed data back to the cache**
- **Each application uses a standard high data-rate interface to a large, high-speed, application-oriented cache**
- **Usually metadata is recorded in a cataloguing system**
- **Usually there is a tertiary storage system manager that migrates data to and from the cache**
  - **the cache can serve as a moving window on the object/dataset (depending on the size of the cache relative to the objects of interest, only part of the object data may be loaded in the cache - though the full objection definition is present)**

Model

- **The native cache access interface is at the logical block level, but client-side libraries implement various access I/O semantics**
  - **E.g. Unix, C I/O:**
    - **upon request available data is returned,**
    - **requests for data in the dataset, but not yet migrated to cache, causes the application-level read to block.**

Generally, the cache is large compared to the available disks of the computing environment, and very large compared to any single disk (e.g. hundreds of gigabytes).

Model



1. Overall Data Acquisition and Processing Model

Prototype Systems

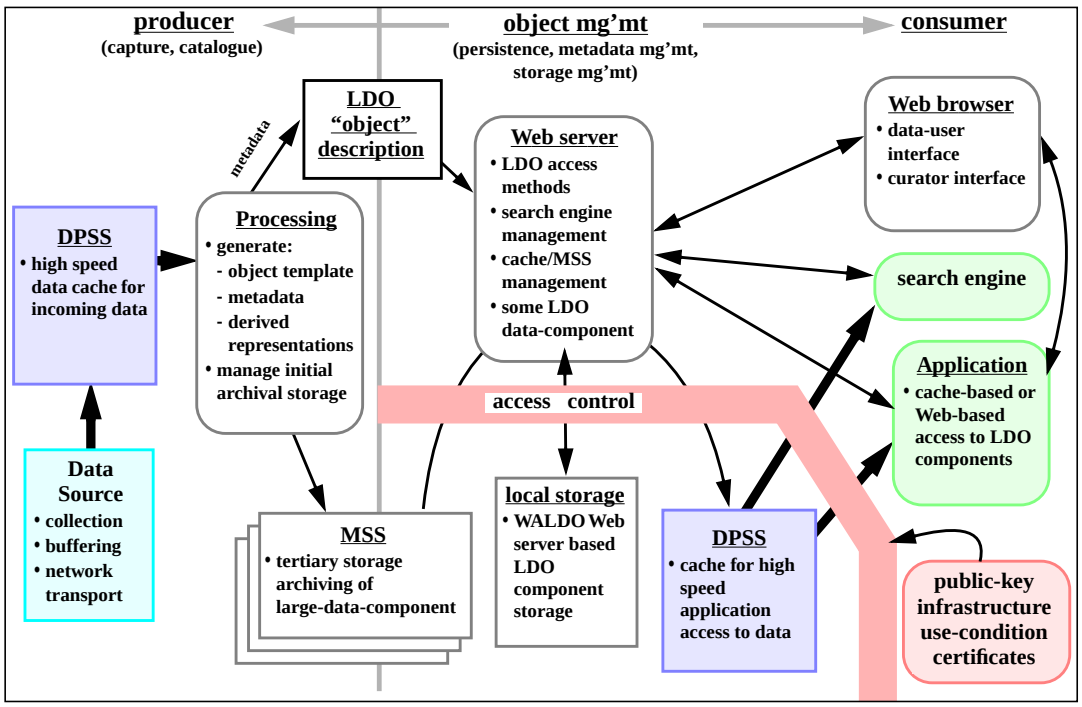
- **WALDO - a system in which real-time data sources are catalogued into a digital library (see http://www-itg.lbl.gov/WALDO)**
- **STAR: High Energy and Nuclear physics accelerator detectors are the prototypical high data-rate scientific instrument, and they have substantial real-time analysis requirements *(discussed in the LBNL - NTON - SLAC experiment)***

Wide-Area, Large Data Object system (WALDO)

- **WALDO was the first system in which we used the DPSS for high-speed data collection and on-line distributed processing of that data.**

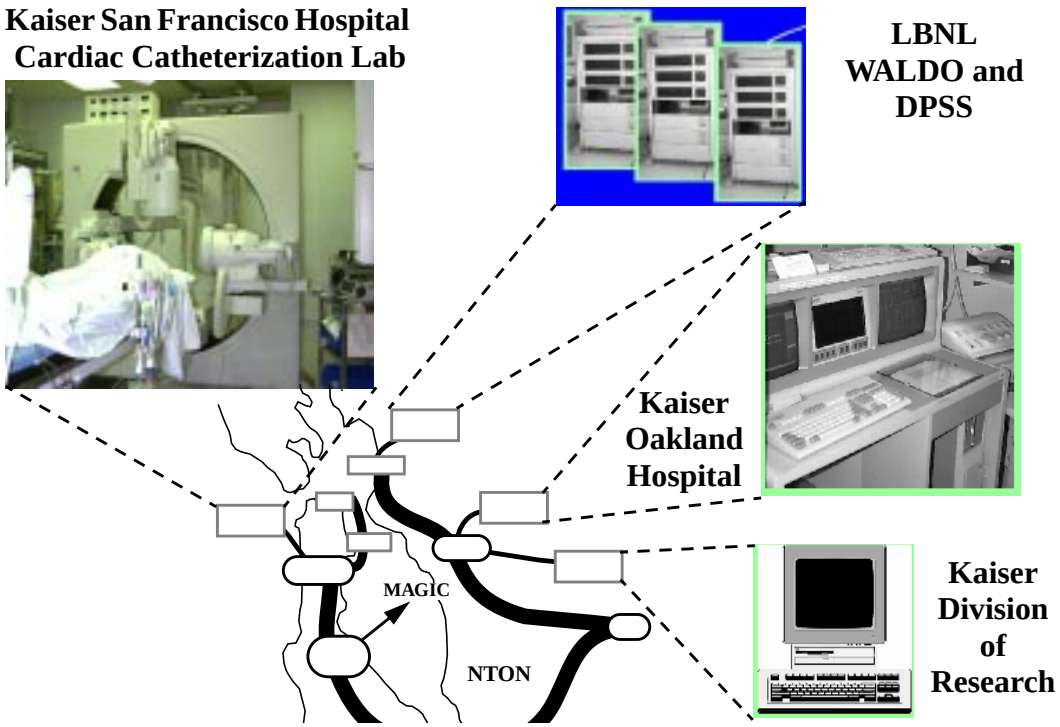
  This general model has been used in several data-intensive computing applications. For example, a real-time digital library system (see figure 2 and [DIGLIB]) collects data from a remote medical imaging system, and automatically processes, catalogues, and archives each data unit together with the derived data and metadata, with the result being a Web-based object representing each dataset. This automatic system operates 10 hours/day, 5-6 days/week with data rates of about 30 Mbits/sec during the data collection phase (about 20 minutes/hour).

WALDO



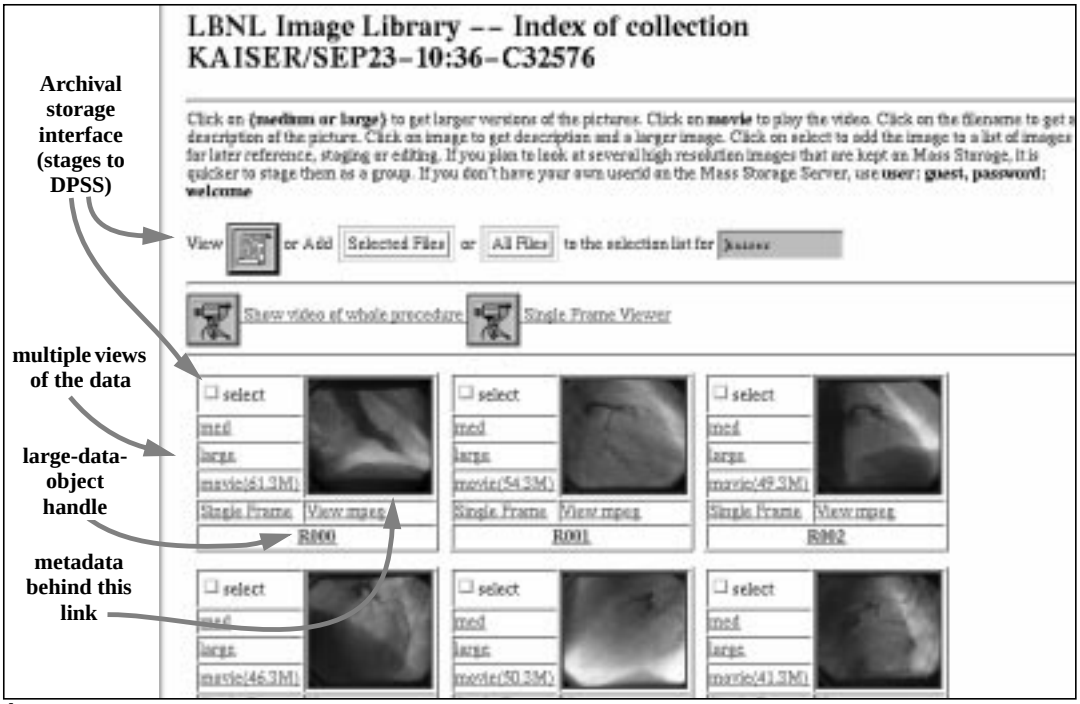
2. WALDO Data Flow for Automatic Digital Library Generation

WALDO



3. Kaiser / LBNL, WALDO based health care imaging system and NTON

WALDO



4. The user view of WALDO: A representation of the six objects (about 0.75 GBy total) resulting from a single cycle of operation of a remote, on-line cardio-angiography system

DPSS: A Wide-Area Network Distributed Cache Architecture

The Distributed-Parallel Storage System (DPSS) is a high-speed, application-oriented network data cache that is itself a widely distributed system, and that serves several roles in high-performance, data-intensive computing environments.

Functionally, the DPSS provides:

- **A standard interface for high-speed data access with the functionality of a single, very large, random access, block oriented I/O device (i.e. a "virtual disk");**
- **High capacity, on-line cache storage (we anticipate a terabyte size for the STAR analysis environment) that serves to isolate the application from the tertiary storage system and the instrument (detector data acquisition system);**

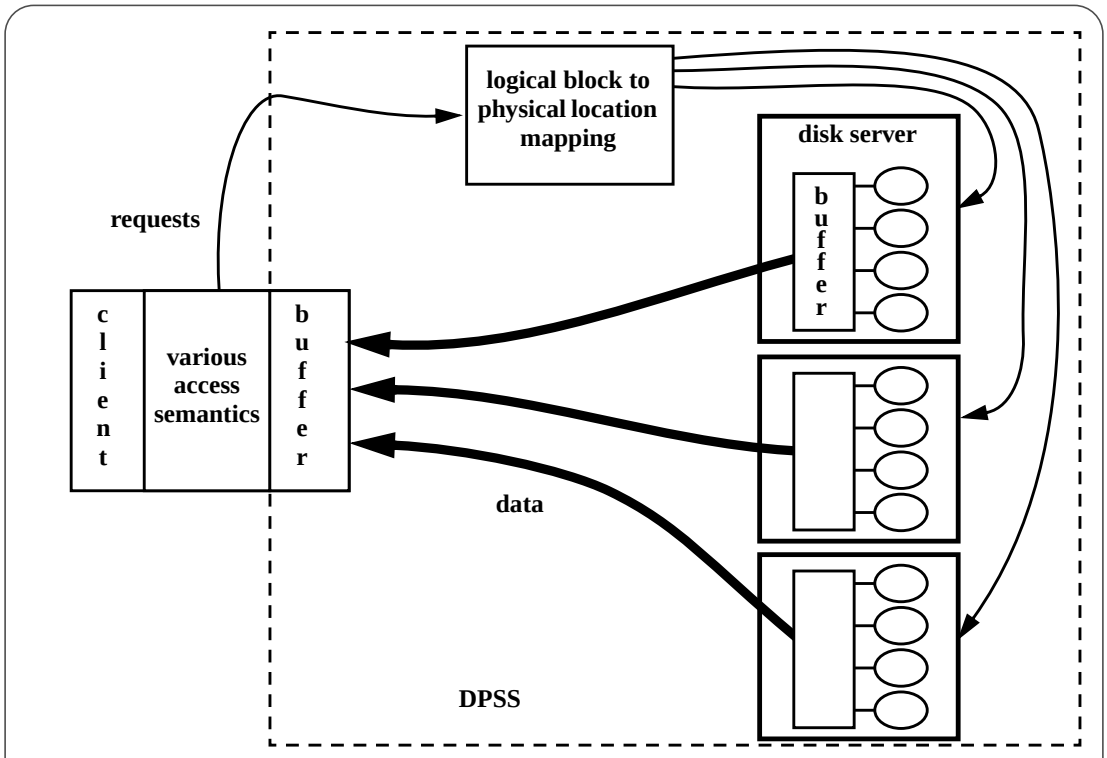
DPSS: A Network Distributed Cache

- **Access to many large datasets that may be logically present in the cache by virtue of the block index maps being loaded, even if the data is not yet available (in this way processing can begin as soon as the first data has been migrated from tertiary storage);**
- **Application-specific interfaces to an extremely large space (16 byte indices) of logical blocks;**
- **The ability to dynamically configure on-line systems by aggregating workstations and disks from all over the network (this is routinely done in the MAGIC testbed);**
- **The ability to build large, high-performance storage systems from the least expensive commodity components;**
- **Scalable performance by increasing the number of parallel operating DPSS servers.**

DPSS: A Network Distributed Cache

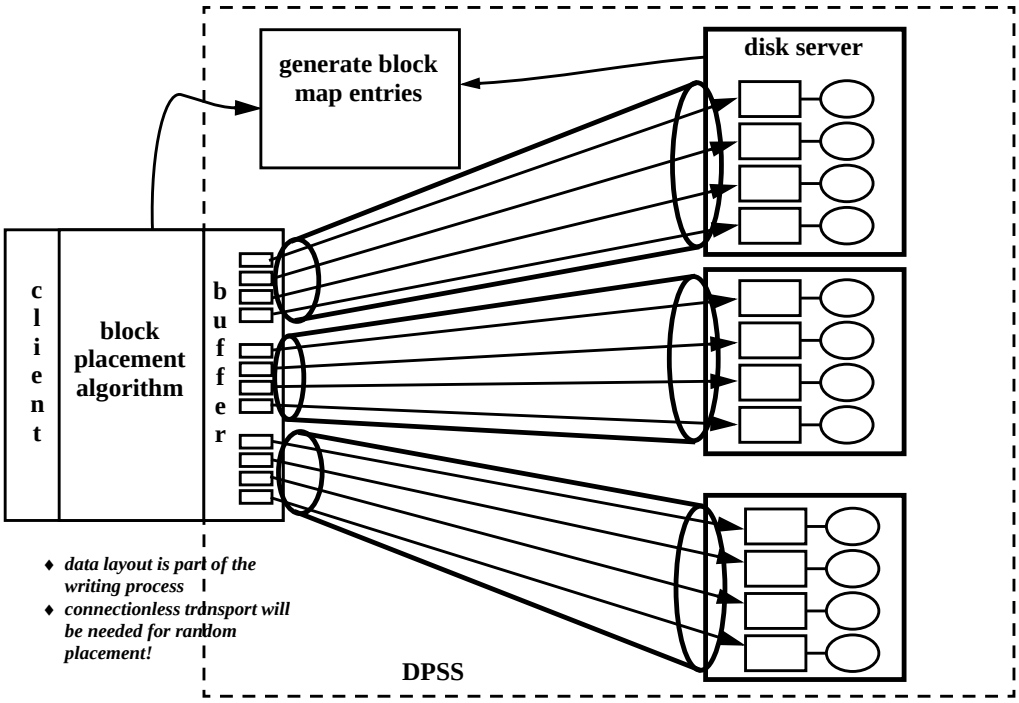
- **DPSS uses parallel operation of distributed servers to supply, for example, image streams fast enough to enable various multi-user, “real-time”, virtual reality-like applications in an Internet / ATM environment.**
- **Ultimately, the ene-to-end performance comes from using separate threads of control for every resource**
- **As illustrated in figure 10. the DPSS provides a “logical block” server that does “third party” transfers from disk servers directly to application client buffers;**
- **The DPSS, as a system, is designed to be distributed across a wide-area network;**
- **The components are actively managed by a collection of independently communicating agents to provide highly distributed, reliable, wide-area operation;**

DPSS: A Network Distributed Cache



8. Distributed-Parallel Storage System Model (Reading)

DPSS: A Network Distributed Cache

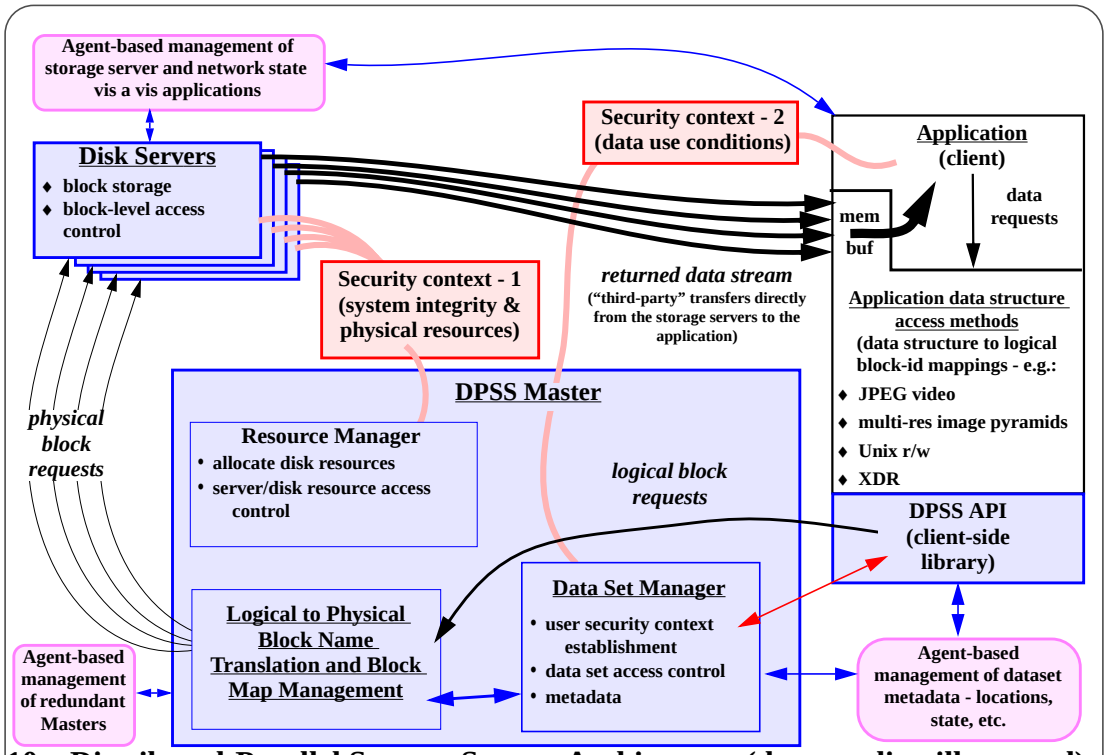


9. DPSS Model for High-Speed Writing

DPSS: A Network Distributed Cache

- **Data placement occurs during the data-write operation (figure 9.)**
- **Three major architectural components: storage and control, security, agent-based management (figure 10.)**

DPSS: A Network Distributed Cache



10. Distributed-Parallel Storage System Architecture (data reading illustrated)

Performance Monitoring and Analysis

There are virtually no behavioral aspects of *widely distributed* applications that can be taken for granted - they are fundamentally different from LAN based distributed applications.

- **A hard problem that is a barrier to the routine construction of high-speed distributed applications**
- **Hard: techniques that work in the LAN frequently do not work in the wide area, and more techniques are needed in the wide area (to address problems that never show up in LANs)**

To characterize the wide area environment we have developed a methodology for detailed, *end-to-end, top-to-bottom monitoring* and analysis of every significant event involved in distributed systems data interchange.

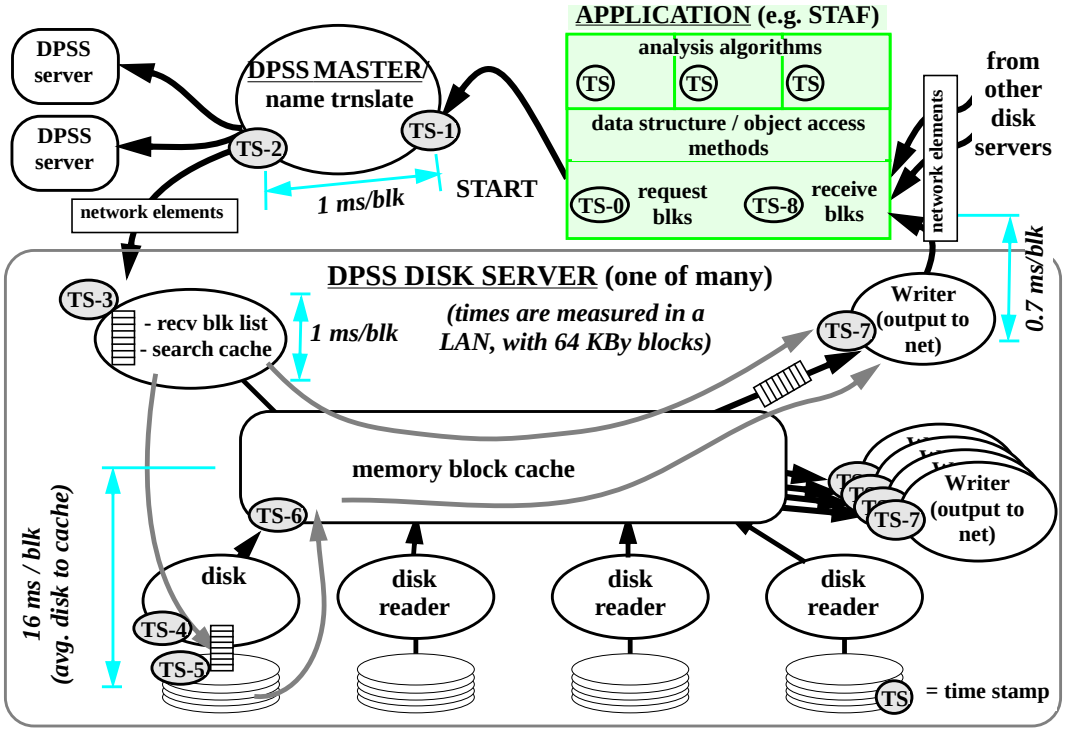
Performance Monitoring and Analysis

- **Has proven invaluable for isolating and correcting performance bottlenecks, and even for debugging distributed parallel code**
- **The monitoring methodology uses precision time correlation of events throughout the distributed system, together with analysis techniques for obtaining information from the reconstructed dataflow lifelines.**
  - **NetLogger/LogTracer tools collect state information and time stamps at all critical points in the data path, including instrumenting the clients and applications**
  - **timing and event information is carried as a defined part of the data block structure OR is logged and correlated based on the timestamps**
  - **NTP synchronizes system clocks to within about 250 microseconds (but the systems have to stay up for a significant length of time for the clocks to converge to 250 μs)**

Performance Monitoring and Analysis

- **The results are detailed, data block transit history "life-lines"**
- **Analysis of these lifelines provides diagnostic information for the entire environment as well as a visualization of how distributed-parallel algorithms operate**

Performance Monitoring and Analysis

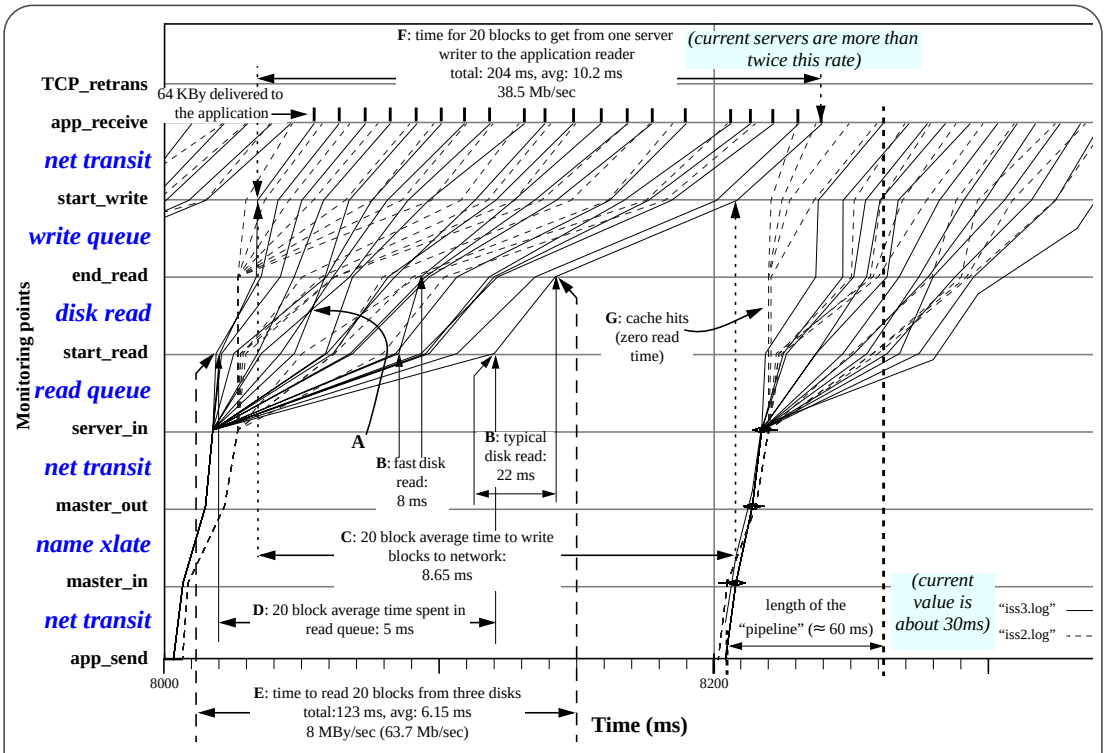


11. End-to-end, top-to-bottom, event monitoring

Performance Monitoring and Analysis

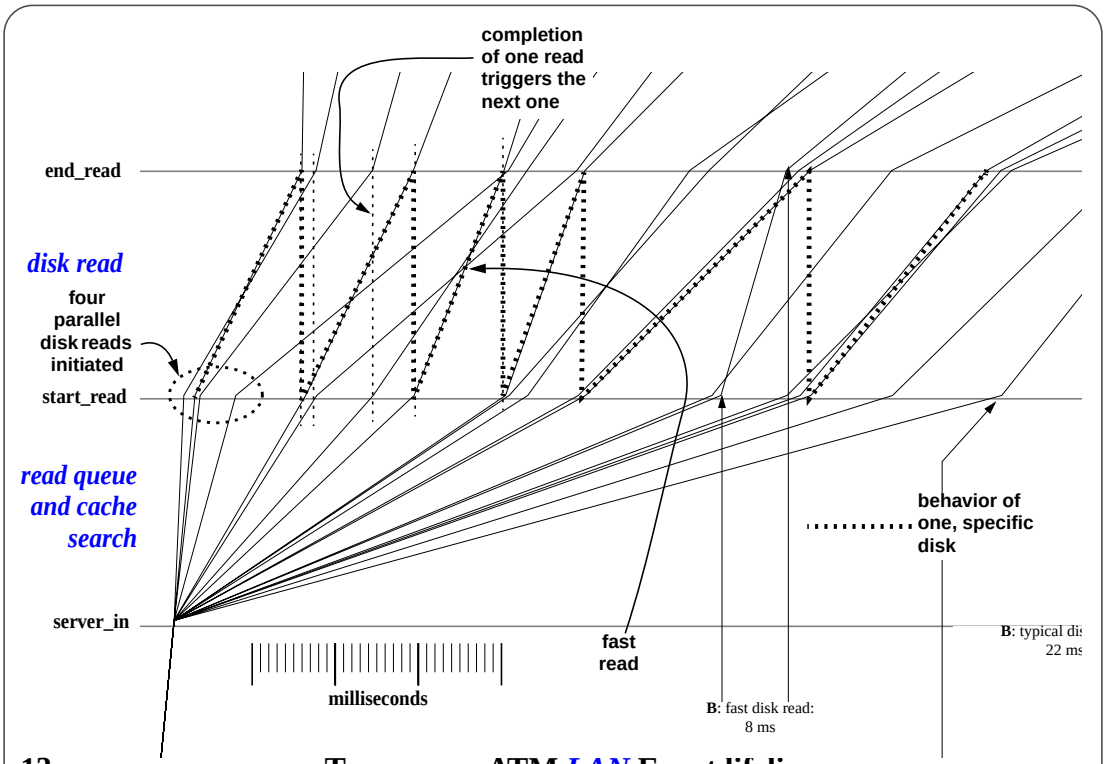
- **Analysis of the event life-lines**
  - **component performance**
  - **algorithm characteristics and interaction**

Performance Monitoring and Analysis



12. Two server, ATM *LAN* Event lifelines (detailed analysis)

Performance Monitoring and Analysis

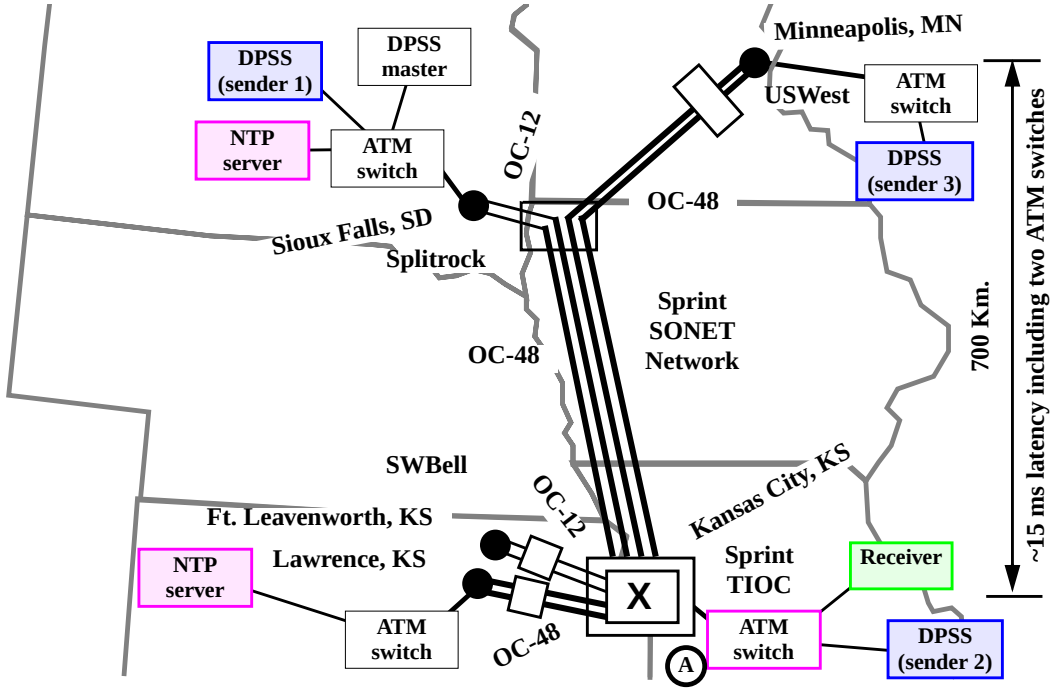


13. Two server, ATM *LAN* Event lifelines

Results of the MAGIC WAN experiment

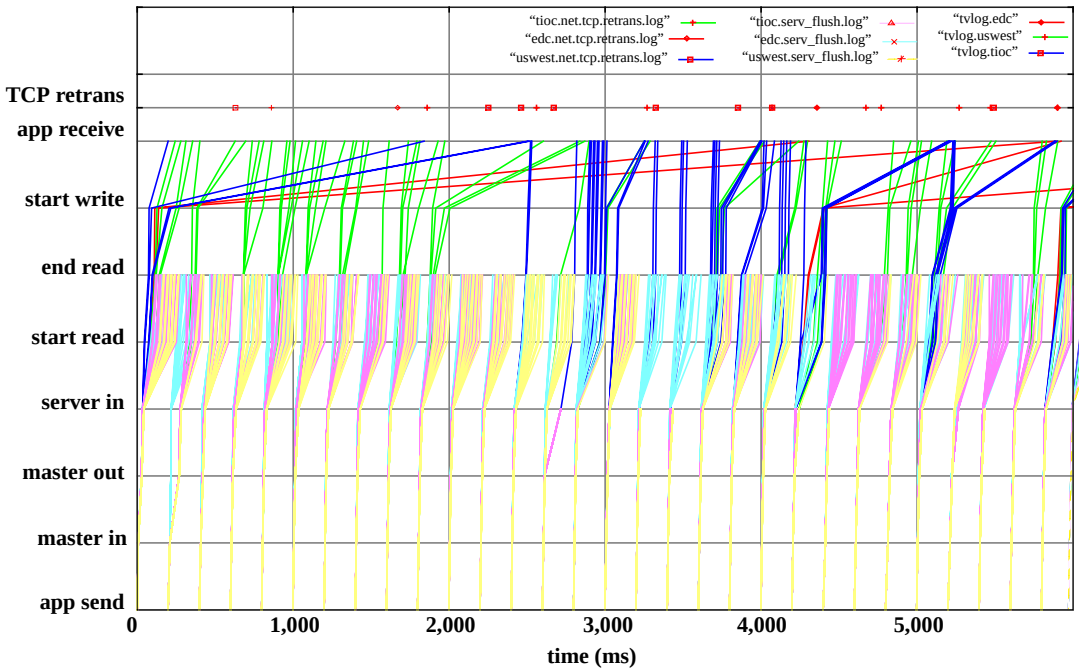
- **Three disk server configuration (next figure) DPSS**
- **Things to notice:**
  - **Many TCP retransmissions, and some very long delays (up to 5500 ms!) (Once a block is written to the TCP socket, the user level flushes have no effect, and TCP will re-send the block until transmission is successful, even though the data is likely no longer needed and is holding up newer data).**
  - **These long delays are almost always accompanied by one or more TCP retransmit events.**

MAGIC WAN Experiment



15. **The MAGIC WAN Network and DPSS / Application Performance Test Configuration**

MAGIC WAN Experiment



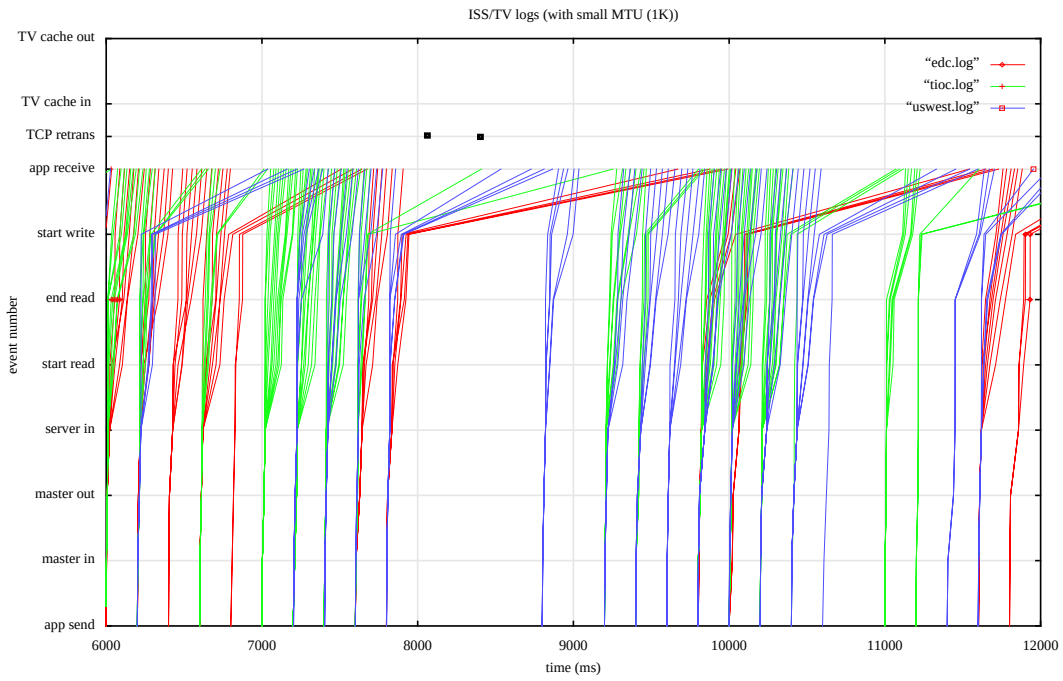
16. Three servers, ATM WAN: things can go very wrong

MAGIC WAN Experiment

Experiment to test TCP window vs. MTU hypothesis: Reduce MTU size

- **Reduce the network MTU to a very small value (e.g. 1024 bytes) so that the TCP window can close to values more consistent with the available switch buffering**
  - **indeed, this helps a lot (see next figure)**

MAGIC WAN Experiment



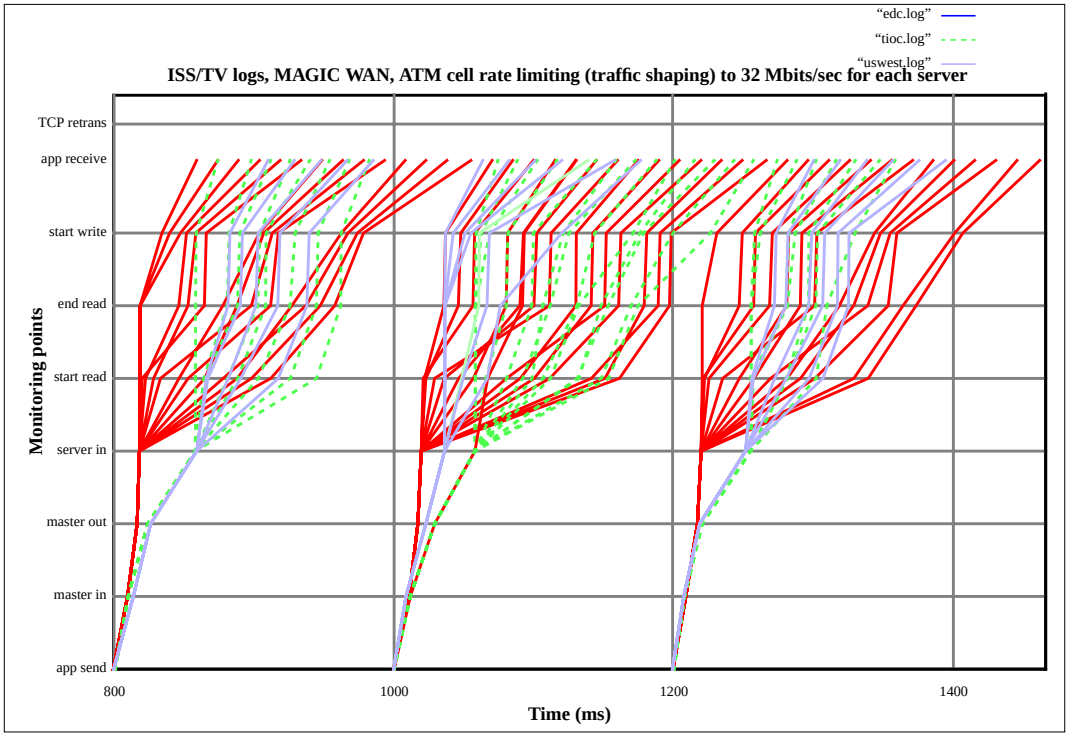
17. Small MTU experiment, MAGIC WAN

MAGIC WAN Experiment

Experiment to test switch cell dropping hypothesis: Cell pacing

These experiments caused all of the MAGIC backbone switches to be replaced and/or upgraded with large output buffers.

MAGIC WAN Experiment



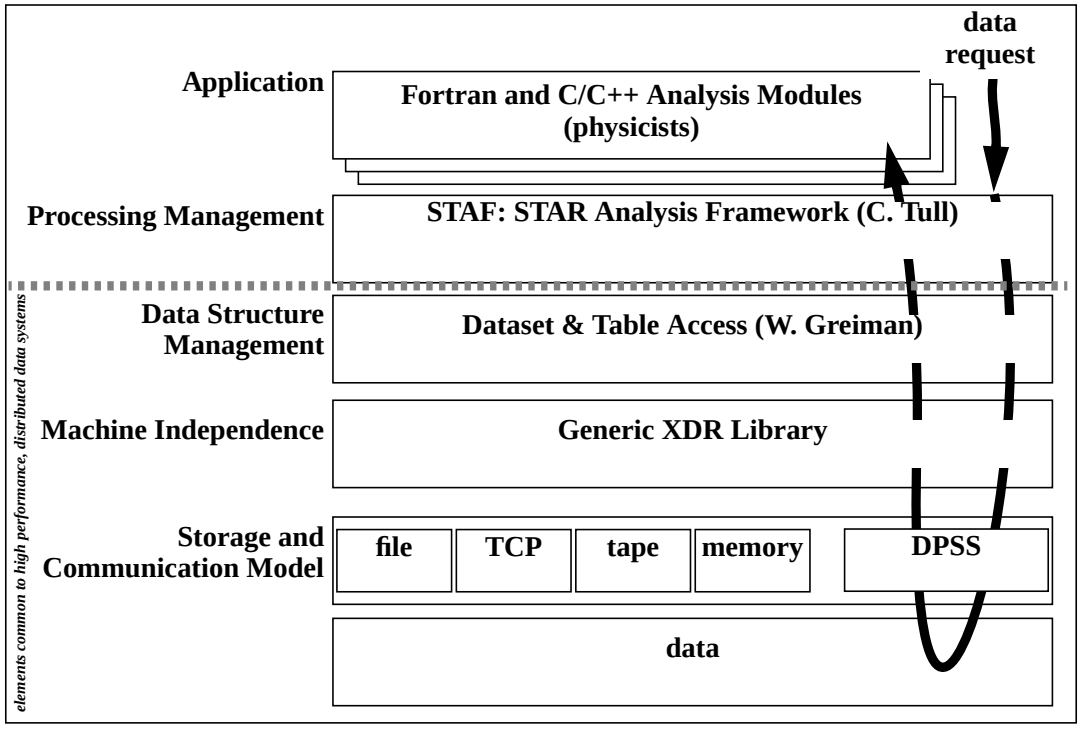
18. The Three Server ATM WAN Experiment with Cell Rate Limiting

Application Performance Monitoring Example

The STAR analysis framework (STAF) is being used to provide a realistic application environment in which to validate and refine the data handling architecture and implementation.

Generally speaking, STAF manages self-describing data structures on behalf of analysis modules. Data is requested through a standard interface that supports several communications models, including the DPSS cache. The data is converted to machine-specific format and placed into memory data structures, whence it is accessed by the analysis modules.

Application Performance Monitoring (STAR Analysis)

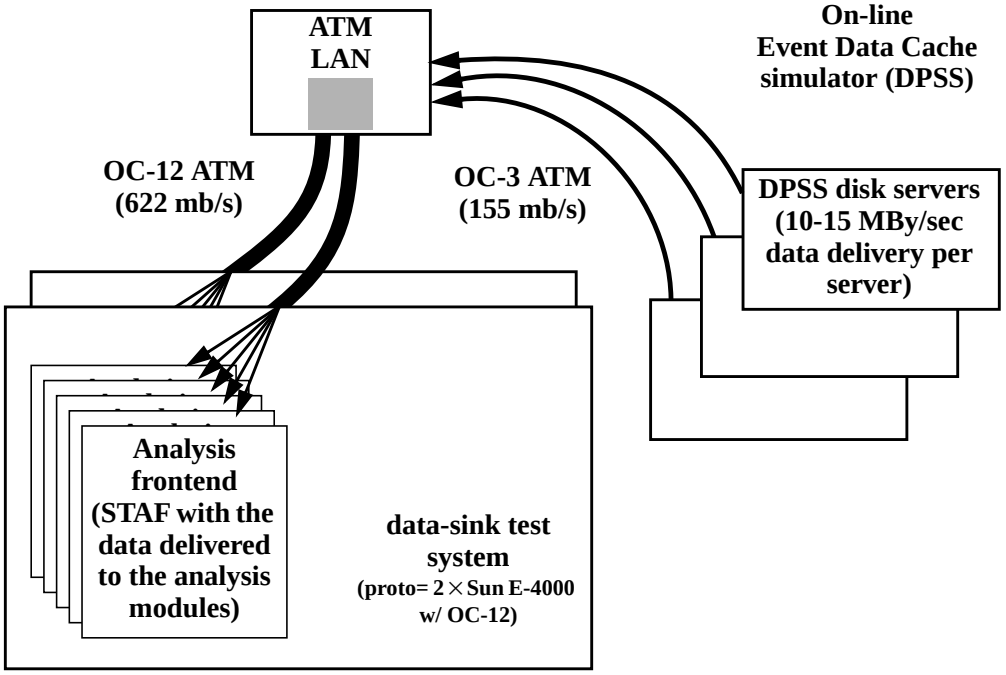


19. Software Architecture for Analysis (e.g. STAR)

Application Performance Monitoring (STAR Analysis)

- A typical DPSS server consists of a commodity workstation (e.g. a 200 MHz Pentium) with one high speed network interface (100 Mb/s Ethernet or 155 Mb/s ATM), three or more SCSI adaptors, and three or more disks on each SCSI string.
- Each such server can independently deliver about 10 Mbytes/sec of data to a remote application which sees the aggregated streams for all servers in a DPSS system.
- Performance in an HENP-like configuration for multiple, parallel applications was measured using a two disk server, four disk, DPSS configuration. Data requests are made by a STAF based reader through the DPSS file semantics interface which collects blocks from the DPSS servers, buffers them, and provides serial access to the buffer through an API.

Application Performance Monitoring (STAR Analysis)



20. DPSS Performance Testing with Many Parallel Applications

Application Performance Monitoring (STAR Analysis)

The throughput rates are measured as data is delivered to the application level (analysis modules) **-**

a data path that includes translating the data to the appropriate machine format and structuring it in memory, both of which are very fast operations -

which provides the most realistic performance measurement.

- **The experiment**
  - **STAF was run on an 8 CPU Sun E-4000 system with an OC-12 (622 Mbit/s) ATM interface**
  - **each instance of the analysis code reads 10 Mbytes of data from each of ten different datasets - a total of 1 Gbyte is read per STAF invocation**

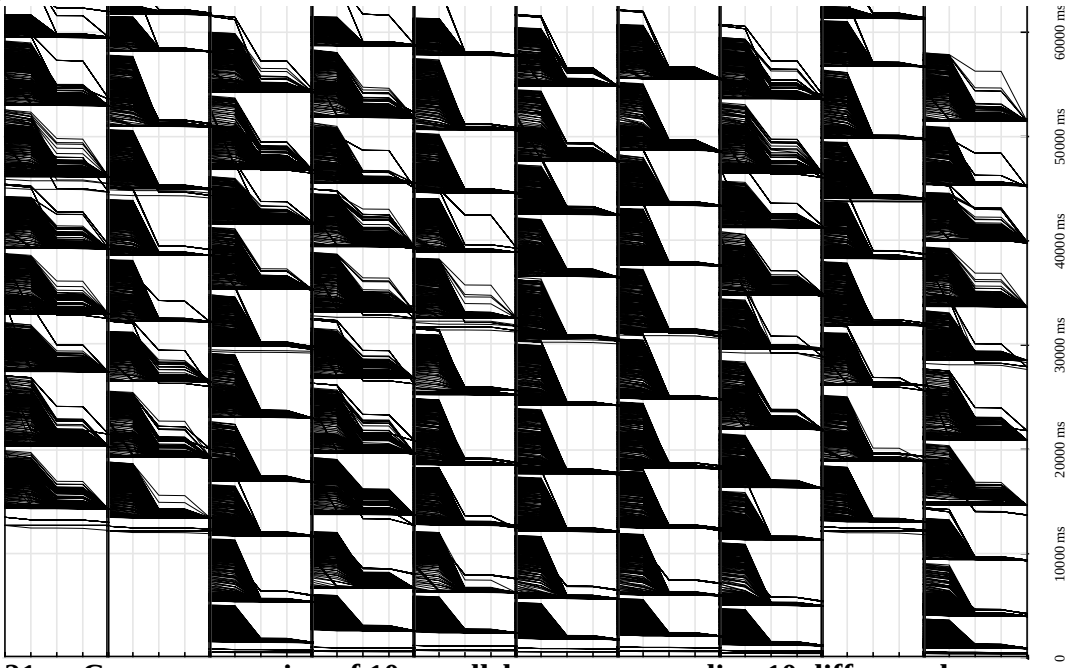
Application Performance Monitoring (STAR Analysis)

- **the DPSS cache was a two server, four disk per server, configuration**
- **a data rate of 23 Mbytes/s is achieved for reading data (as expected for two disk servers), and 25 MBytes/s for writing data.**

♦ **Result (next figure)**

- **running 10 instances of the application simultaneously results in the same aggregate throughput with very uniform access, indicating good scalability.**

Application Performance Monitoring (STAR Analysis)



21. Correct operation of 10 parallel processes reading 10 different data sets from one DPSS (each row is one process, each group is a request for 10 Mbytes of data, total = 1 GBy, for *1.5 TBy/day* - the detector output rate)

The LBNL-SLAC-NTON High Data-Rate Experiments

- Brian Tierney[1,2], Jason Lee[2], Craig Tull[3], and Bill Johnston[1], LBNL
- Les Cottrell[2] and Dave Millsom[2], SLAC
- Bill Lennon[2] and Lee Thombley[2], LLNL/NTON
- Hal Edwards[2], Nortel

[1]middleware and monitoring
[2]networks
[3]applications

An Experiment in High-Speed, Wide Area Real-Time Distributed Data Handling: The STAR analysis environment.

- A four-server DPSS is a prototype frontend for a high-speed mass storage system, and is located at LBNL.
- A 4 CPU Sun E-4000 is a prototype for a physics data analysis computing cluster, and is located at SLAC.
- The NTON network testbed that connects LBNL and SLAC provides a five switch, 100 km, OC-12 path (and can be configured for a 2000 km, OC-12, path).
- All experiments are application-to-application, using TCP transport

Wide Area Distributed Data Handling

Questions to be addressed:

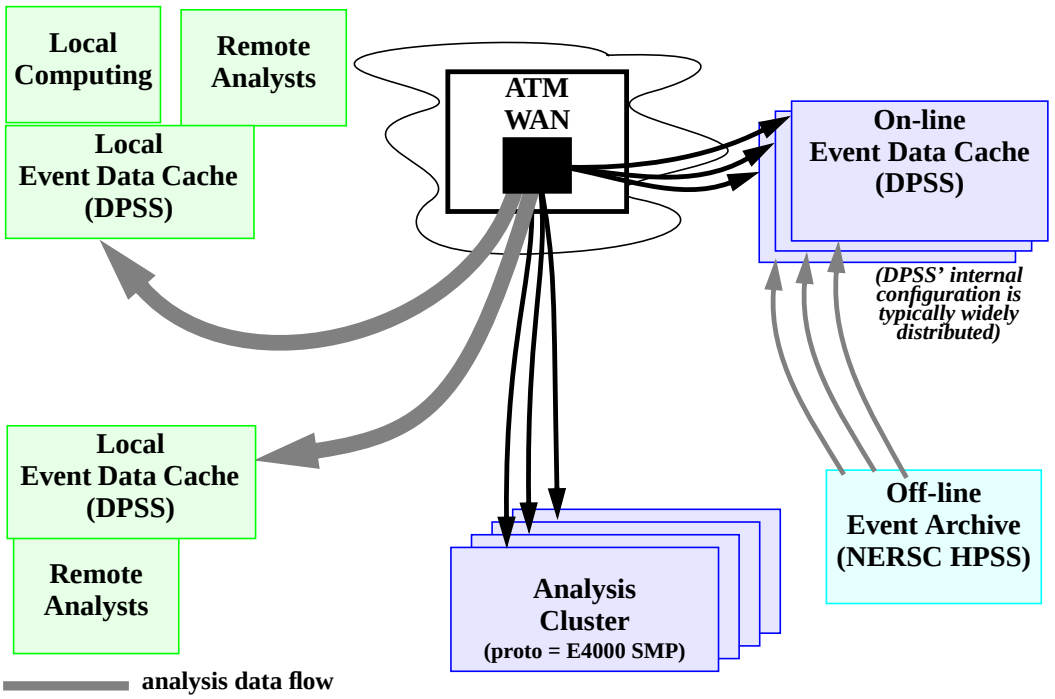
- **What are the issues with the routine use of remote network disks to support high data-rate environments?**
  - **Are there any additional issues if the network cache is the frontend for a high performance mass storage system (HPSS)?**
  - **How well does TCP congestion control work for high-speed data streams in high-speed networks?**
  - **How do we do high-speed rate shaping for QoS?**
- **How do anomalies in a separate control channel connection impact performance (as might be encountered if control information were going over a ground line while data was going over a high bandwidth-delay product link like ACTS).**
- **How does striping across multiple, independent physical paths impact performance in a parallel-distributed system?**

Wide Area Distributed Data Handling

Approach: monitoring data flow related events top-to-bottom, and end-to-end

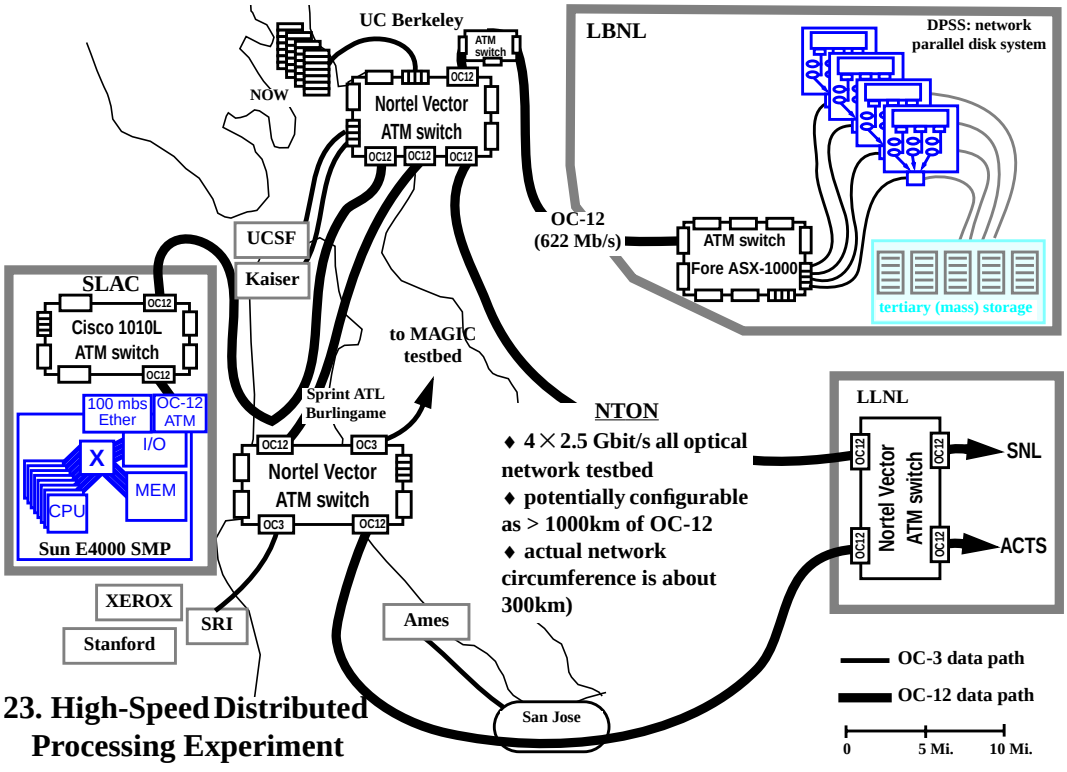
- **Application software architecture is that of the STAR analysis framework**
- **Network architecture**
- **Application-level results**

Wide Area Distributed Data Handling



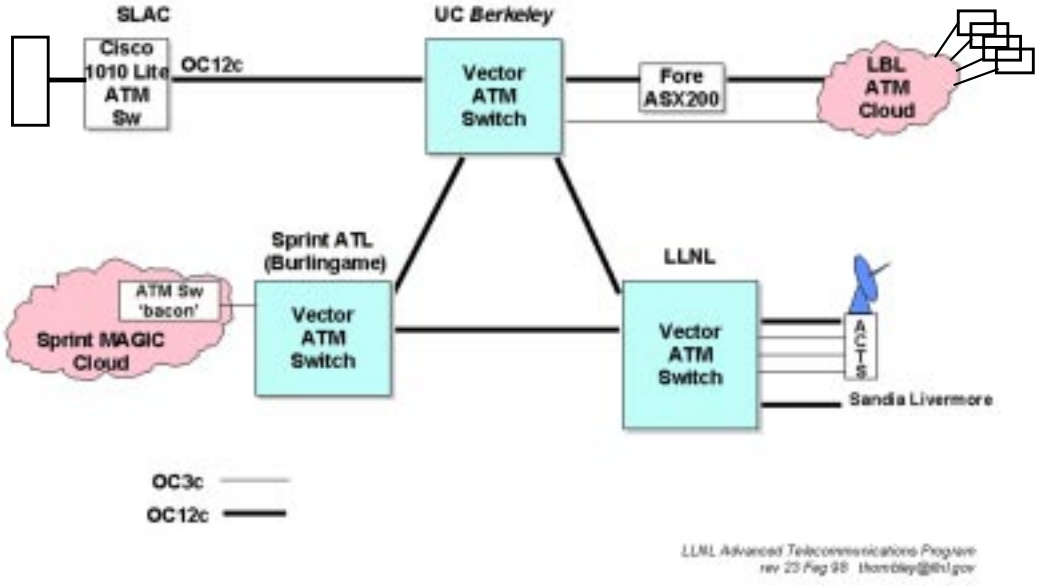
22. Prototype Analysis Architecture

Wide Area Distributed Data Handling



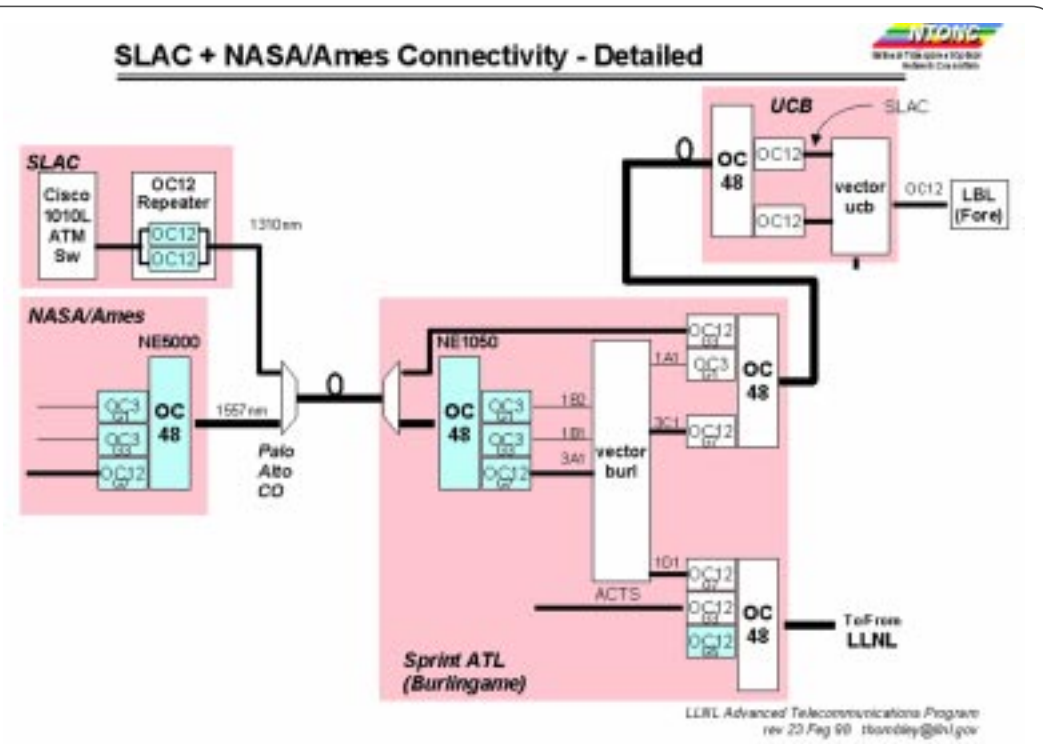
23. High-Speed Distributed Processing Experiment

Wide Area Distributed Data Handling



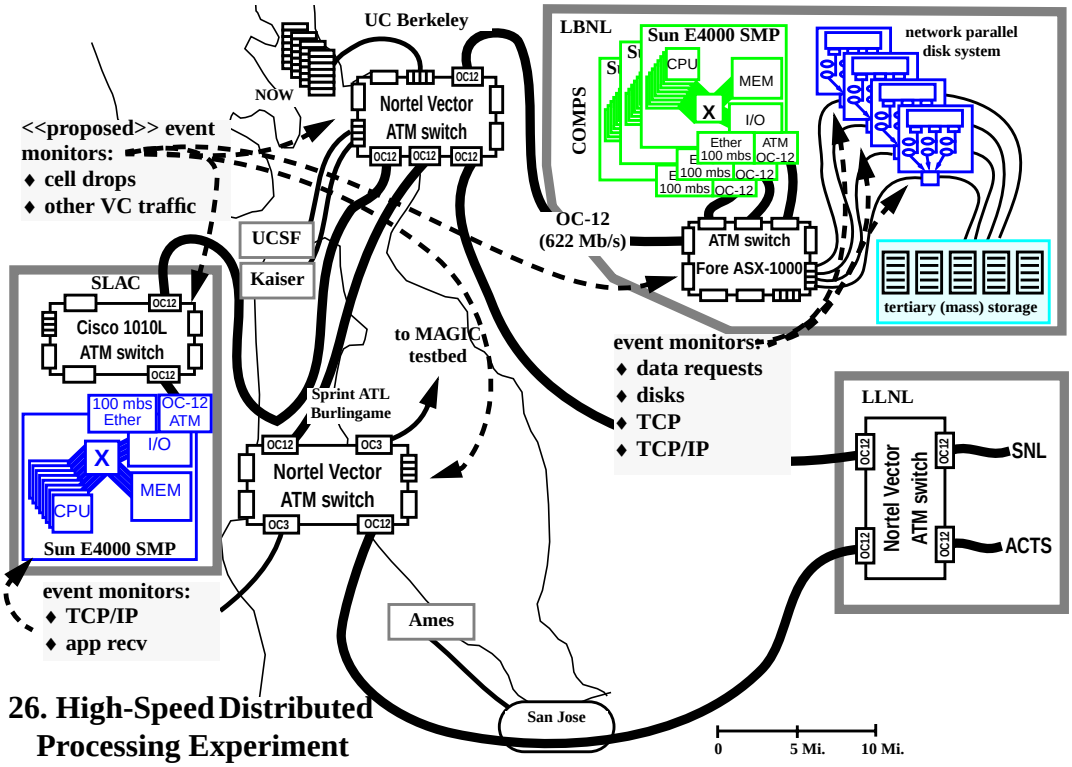
24.

Wide Area Distributed Data Handling



25.

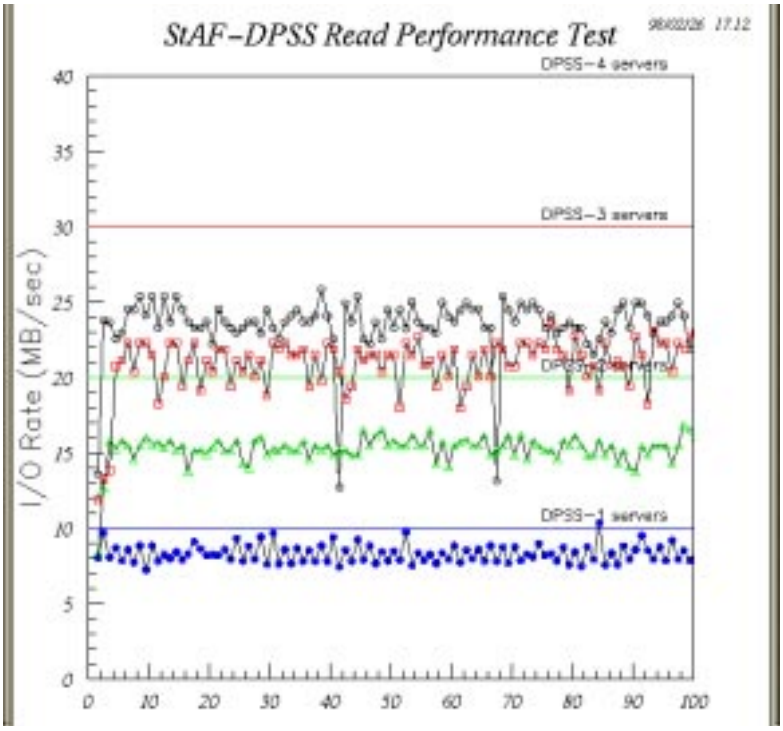
Wide Area Distributed Data Handling



26. High-Speed Distributed Processing Experiment

Wide Area Distributed Data Handling

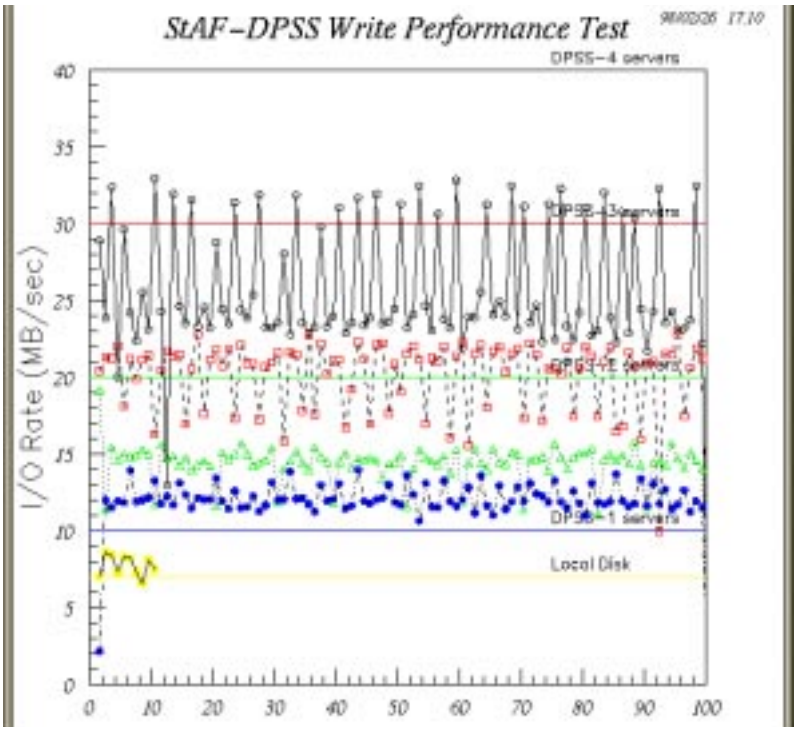
Rates of data received in the application's memory (includes XRD decode)



27. STAF

Wide Area Distributed Data Handling

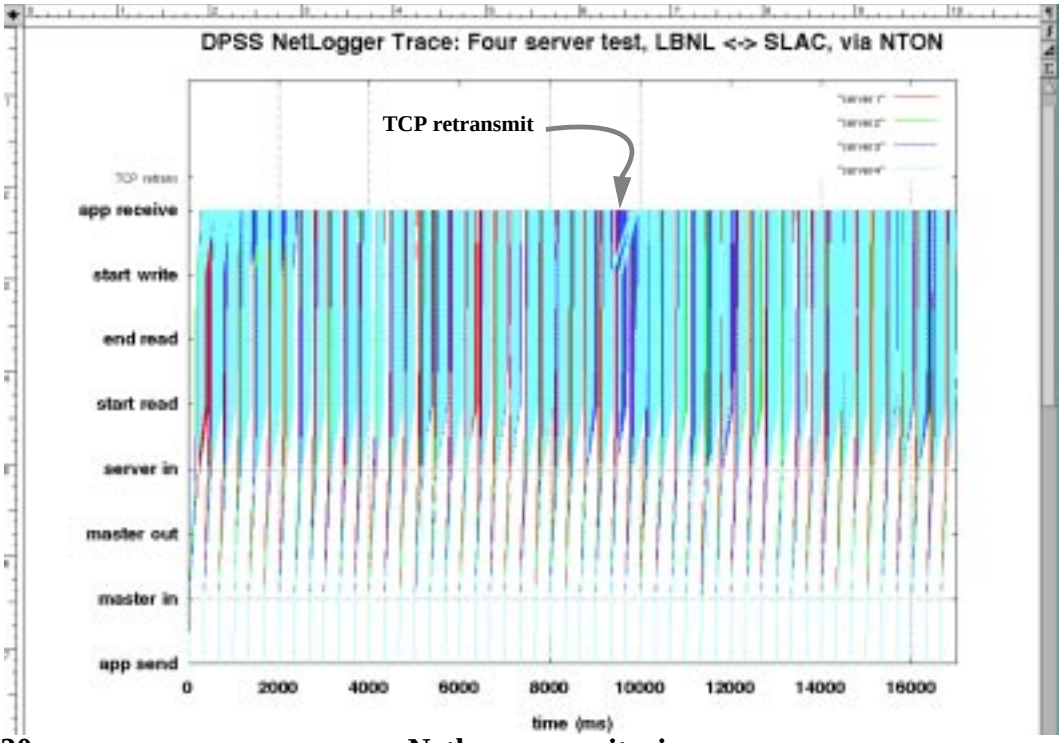
Rates of data written out of the application's memory ***(note comparison of a 1-server DPSS with local disk)***



28. **STAF**

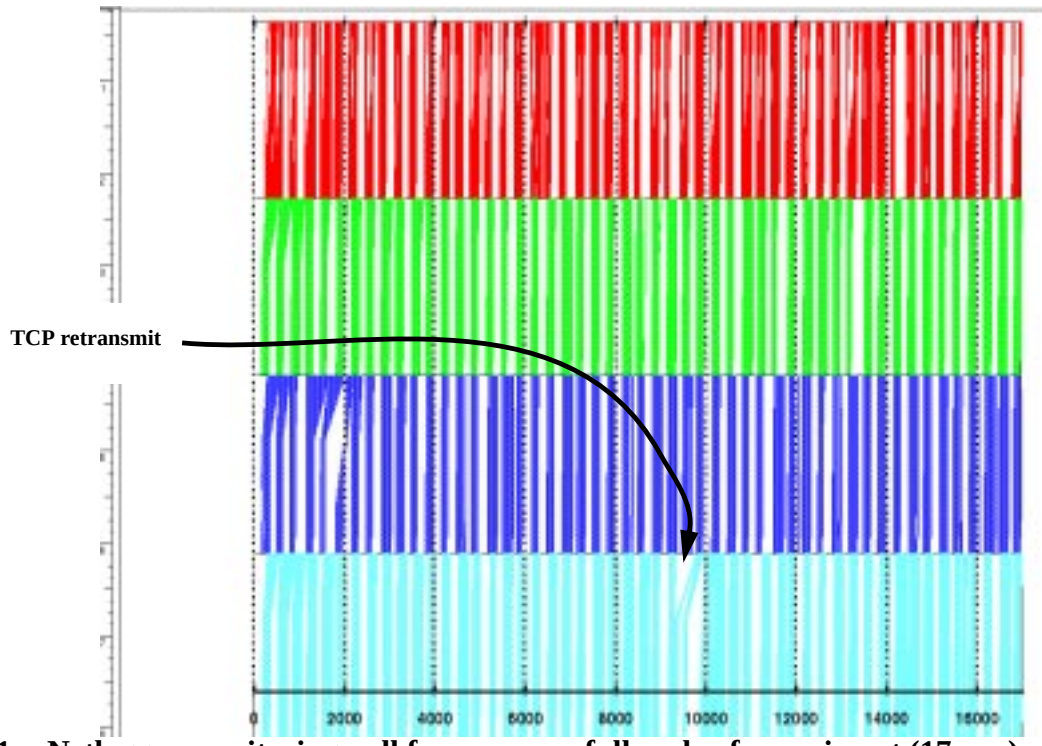
Wide Area Distributed Data Handling

Netlogger monitoring



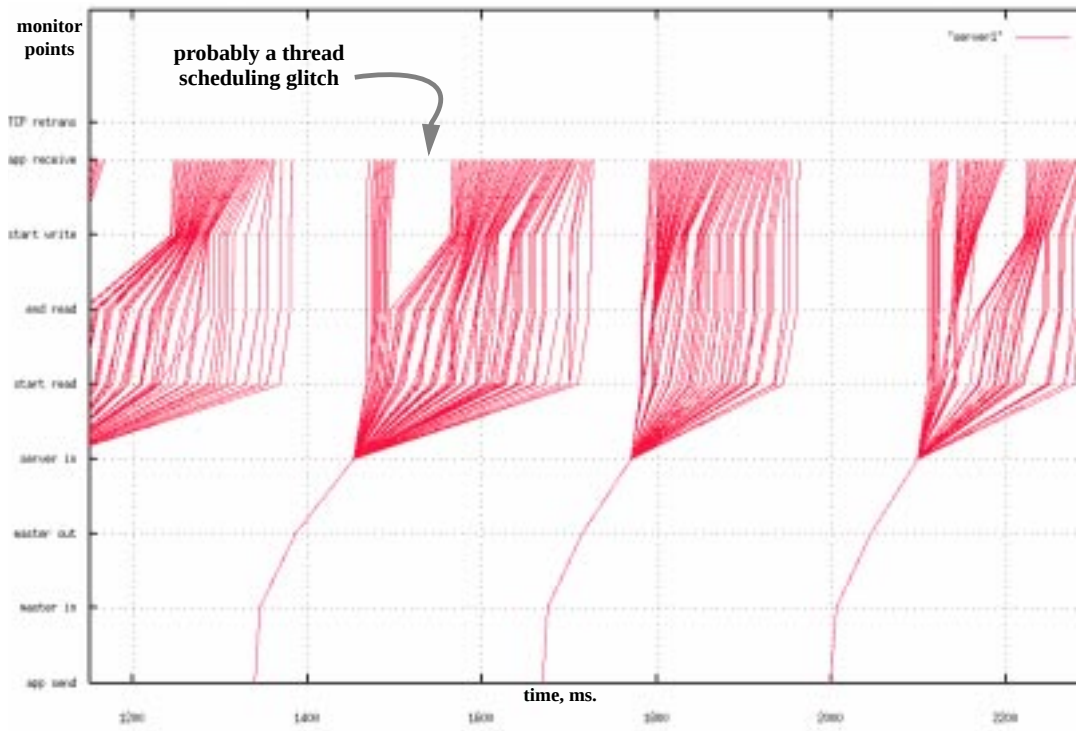
30. Netlogger monitoring

Wide Area Distributed Data Handling



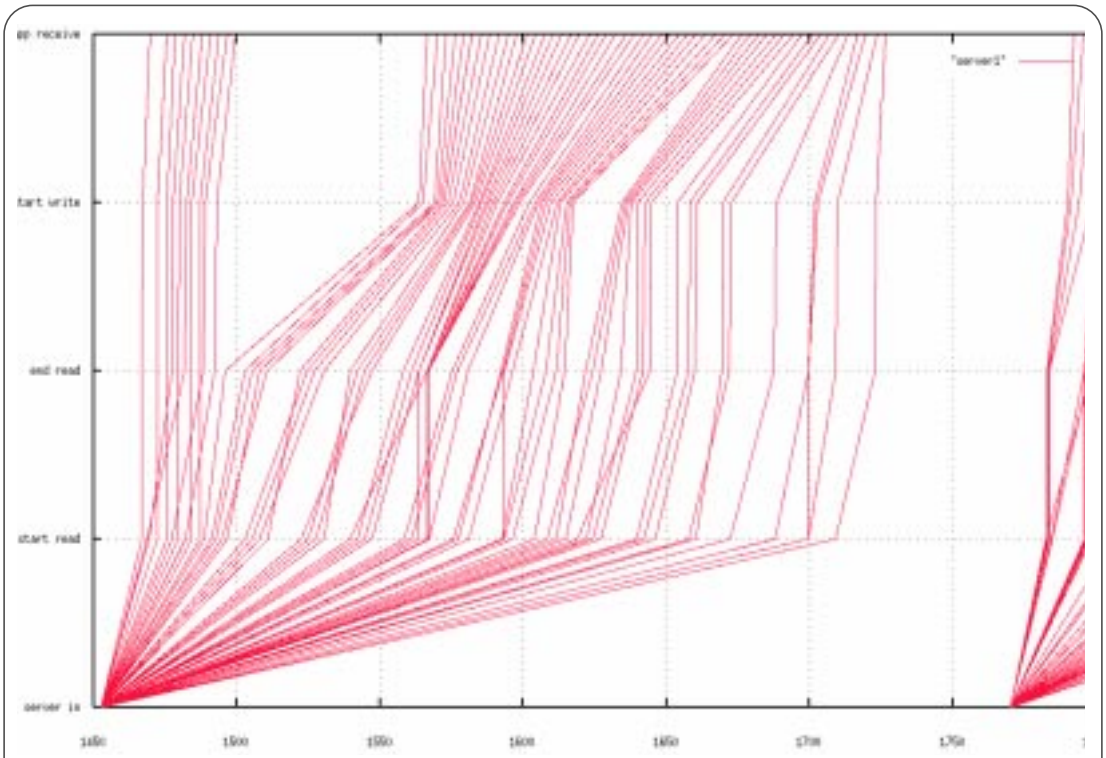
31. Netlogger monitoring - all four servers, full cycle of experiment (17 sec.)

Wide Area Distributed Data Handling



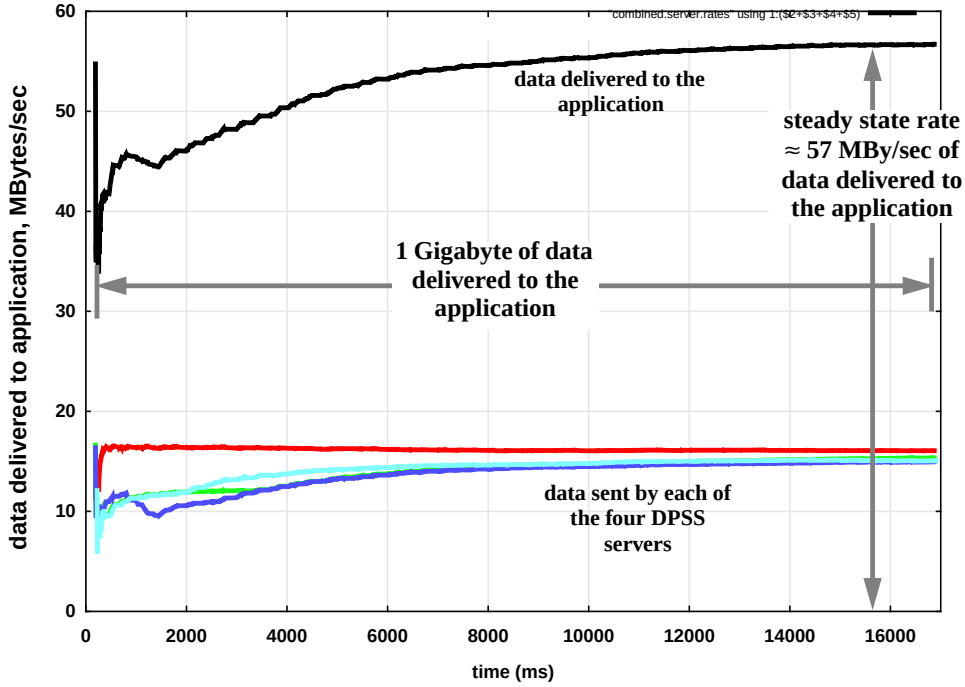
32. Netlogger monitoring - detail analysis example

Wide Area Distributed Data Handling



33. Netlogger monitoring

DPSS NetLogger Trace: Four server test, LBNL <-> SLAC, via TCP over NTON



29. Experiment: Application receiving data from four remote servers.

Conclusions

The experiments described here are work-in-progress. The use of the Distributed Parallel Storage System as a network cache has demonstrated the required performance, but a complete demonstration of scalability requires running hundreds of analysis processes along with other, independent uses of the infrastructure.

The wide area, high-data rate experiment configuration results are described here, and meet our expectation of being able to deliver data to applications at rates limited only by the network and the ability of the application and its platform to consume data (i.e, the DPSS architecture is thus far scalable). However, experience has also shown that every significant increase in throughput and/or scale raises a new set of issues.

We also report on the Netlogger monitoring system which has been essential for debugging and verifying high-speed distributed applications and the supporting infrastructure.

There are numerous issues that remain to be investigated, identified, and addressed. For example, the behavior of the infrastructure with competing, high-speed applications, coupling the monitoring architecture to the middleware and applications so that adaptations can be made for performance degradation in the various components, etc.

References and Notes

[DIGLIB] "Real-Time Generation and Cataloguing of Large Data-Objects in Widely Distributed Environments", W.Johnston, Jin G., C. Larsen, J. Lee, G. Hoo, M. Thompson, B. Tierney, J. Terdiman. To be published in International Journal of Digital Libraries - Special Issue on "Digital Libraries in Medicine". (Available at http://www-itg.lbl.gov/WALDO)

DOE2000 See http://www-itg.lbl.gov/DCEE

DPSS "The Distributed-Parallel Storage System (DPSS)". See http://www-itg.lbl.gov/DPSS.

Lau94 "TerraVision: a Terrain Visualization System". S. Lau, Y. Leclerc, Technical Note 540, SRI International, Menlo Park, CA, Mar. 1994. Also see: http://www.ai.sri.com/~magic/terravision.html .

MAGIC "The MAGIC Gigabit Network" (http://www.magic.net/)

NTONC "National Transparent Optical Network Consortium". See http://www.ntonc.org . (NTONC is a program of collaborative research, deployment and demonstration of an all-optical open testbed communications network.)

STAR1 "Relativistic Nuclear Collisions Program", H.G. Ritter. http://www-library.lbl.gov/docs/LBNL/397/64/Overviews/RNC.html

STAR2 "High Speed Distributed Data Handling for HENP", W. Greiman, W. E. Johnston, C. McParland, D. Olson, B. Tierney, C. Tull. http://www-rnc.lbl.gov/computing/ldrd_fy97/henpdata.htm

Thomp97 "Distributed Health Care Imaging Information Systems". M. Thompson, W. Johnston, G. Jin, J. Lee, B. Tierney, Lawrence Berkeley National Laboratory, Berkeley CA, and Terdiman, J. F., Kaiser Permanente, Division of Research, Oakland CA. SPIE International Symposium on Medical Imaging, 1997. Newport Beach, California. (Also available at http://www-itg.lbl.gov/Kaiser.IMG/homepage.html)

Tull97 "The STAR Analysis Framework Component Software in a Real-World Physics Experiment". C.Tull, W.Greiman, D.Olson, D.Prindle, H.Ward, International Conference on Computing in High Energy Physics, Berlin, Germany, April, 1997.

[Tierney] "Performance Analysis in High-Speed Wide Area ATM Networks: Top-to-bottom end-to-end Monitoring", B. Tierney, W. Johnston, J. Lee, G. Hoo. IEEE Networking, May 1996.

- **http://www-itg.lbl.gov/DPSS**